

COL724 Report

Chaitanya, 2019CS50435

1. The below screenshot is for the learning switch and controller hub

```
(env) ubuntu@ip-172-31-1-0:~/col724$ sudo python3.8 topology.py
mininet> h1 ping h4
PING 10.0.0.8 (10.0.0.8) 56(84) bytes of data.
64 bytes from 10.0.0.8: icmp_seq=1 ttl=64 time=39.5 ms
64 bytes from 10.0.0.8: icmp_seq=1 ttl=64 time=40.3 ms (DUP!)
64 bytes from 10.0.0.8: icmp_seq=1 ttl=64 time=42.0 ms (DUP!)
64 bytes from 10.0.0.8: icmp_seq=1 ttl=64 time=42.7 ms (DUP!)
64 bytes from 10.0.0.8: icmp_seq=2 ttl=64 time=0.056 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=64 time=0.032 ms
64 bytes from 10.0.0.8: icmp_seq=4 ttl=64 time=0.055 ms
64 bytes from 10.0.0.8: icmp_seq=5 ttl=64 time=0.048 ms
64 bytes from 10.0.0.8: icmp_seq=6 ttl=64 time=0.035 ms
64 bytes from 10.0.0.8: icmp_seq=7 ttl=64 time=0.050 ms
64 bytes from 10.0.0.8: icmp_seq=8 ttl=64 time=0.050 ms
^C
--- 10.0.0.8 ping statistics ---
8 packets transmitted, 8 received, +3 duplicates, 0% packet loss, time 7139ms
rtt min/avg/max/mdev = 0.032/14.985/42.708/19.777 ms
```

```
(env) ubuntu@ip-172-31-1-0:~/col724$ sudo python3.8 topology.py
mininet> h1 ping h4
PING 10.0.0.8 (10.0.0.8) 56(84) bytes of data.
64 bytes from 10.0.0.8: icmp_seq=1 ttl=64 time=9.23 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=64 time=4.14 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=64 time=4.02 ms
64 bytes from 10.0.0.8: icmp_seq=4 ttl=64 time=4.16 ms
64 bytes from 10.0.0.8: icmp_seq=5 ttl=64 time=4.17 ms
64 bytes from 10.0.0.8: icmp_seq=6 ttl=64 time=4.07 ms
64 bytes from 10.0.0.8: icmp_seq=7 ttl=64 time=4.07 ms
64 bytes from 10.0.0.8: icmp_seq=8 ttl=64 time=4.13 ms
^C
--- 10.0.0.8 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7010ms
rtt min/avg/max/mdev = 4.016/4.747/9.231/1.695 ms
mininet>
```

The difference between controller hub and learning switch is
Hub Controller:

- Broadcasts incoming packets to all ports.
- Highly inefficient and causes network congestion.
- Does not perform MAC address learning.
- Rarely used in modern networks.

Learning Switch:

- Forwards packets to specific ports based on MAC address.
- Efficient and reduces unnecessary traffic.
- Learns MAC address-to-port mappings.
- Widely used in modern Ethernet networks for better performance.

```
mininet> dpctl dump-flows
*** s1 ***
cookie=0x0, duration=1250.368s, table=0, n_packets=169, n_bytes=12822, priority=0 actions=CONTROLLER:65535
*** s2 ***
cookie=0x0, duration=1250.370s, table=0, n_packets=169, n_bytes=12822, priority=0 actions=CONTROLLER:65535
mininet>
```

2. Pingall in this case blocks the communication between h1-h4 , h2-h5, h3-h5. Also,

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5
h2 -> h1 h3 h4 h5
h3 -> h1 h2 h4 h5
h4 -> X h2 h3 h5
h5 -> X X X h4
*** Results: 20% dropped (16/20 received)
```

Ways to minimize firewall rules:

- Aggregation of similar rules.
- Default actions for common behaviors.
- Wildcard matches for broader traffic coverage.
- Prioritize specific rules with higher priorities.
- Use multiple tables for efficient rule processing.
- Implement stateful firewalls.
- Automation for dynamic rule management.

The rules can be dumped by `dpctl dump-flows`

```
h3"
cookie=0x0, duration=273.347s, table=0, n_packets=9, n_bytes=714, priority=3, in_port="s1-eth3", dl_dst=a6:6e:44:09:9f:1a actions=output:"s1-eth2"
h2"
cookie=0x0, duration=273.344s, table=0, n_packets=8, n_bytes=616, priority=3, in_port="s1-eth2", dl_dst=e6:1b:6a:a2:38:2e actions=output:"s1-eth3"
h3"
cookie=0x0, duration=273.334s, table=0, n_packets=18, n_bytes=1372, priority=3, in_port="s1-eth4", dl_dst=a6:6e:44:09:9f:1a actions=output:"s1-eth2"
cookie=0x0, duration=273.333s, table=0, n_packets=12, n_bytes=1008, priority=3, in_port="s1-eth2", dl_dst=e4:f8:79:ed:0d:c3 actions=output:"s1-eth4"
cookie=0x0, duration=273.311s, table=0, n_packets=18, n_bytes=1372, priority=3, in_port="s1-eth4", dl_dst=e6:1b:6a:a2:38:2e actions=output:"s1-eth3"
cookie=0x0, duration=273.306s, table=0, n_packets=12, n_bytes=952, priority=3, in_port="s1-eth3", dl_dst=e4:f8:79:ed:0d:c3 actions=output:"s1-eth4"
cookie=0x0, duration=258.272s, table=0, n_packets=4, n_bytes=224, priority=3, in_port="s1-eth2", dl_dst=e6:7f:46:46:04:15 actions=output:"s1-eth4"
cookie=0x0, duration=248.032s, table=0, n_packets=3, n_bytes=238, priority=3, in_port="s1-eth3", dl_dst=e6:7f:46:46:04:15 actions=output:"s1-eth4"
cookie=0x0, duration=207.584s, table=0, n_packets=10, n_bytes=532, priority=2, dl_src=e0:98:26:2b:a6:a5, dl_dst=e4:f8:79:ed:0d:c3 actions=drop
cookie=0x0, duration=291.449s, table=0, n_packets=71, n_bytes=4806, priority=0 actions=CONTROLLER:65535
*** s2 ***
cookie=0x0, duration=287.083s, table=0, n_packets=15, n_bytes=1190, priority=3, in_port="s2-eth2", dl_dst=e0:98:26:2b:a6:a5 actions=output:"s2-eth3"
cookie=0x0, duration=287.080s, table=0, n_packets=21, n_bytes=1498, priority=3, in_port="s2-eth3", dl_dst=e6:7f:46:46:04:15 actions=output:"s2-eth2"
cookie=0x0, duration=273.344s, table=0, n_packets=11, n_bytes=910, priority=3, in_port="s2-eth1", dl_dst=a6:6e:44:09:9f:1a actions=output:"s2-eth3"
cookie=0x0, duration=273.343s, table=0, n_packets=7, n_bytes=462, priority=3, in_port="s2-eth2", dl_dst=a6:6e:44:09:9f:1a actions=output:"s2-eth3"
cookie=0x0, duration=273.333s, table=0, n_packets=25, n_bytes=2058, priority=3, in_port="s2-eth3", dl_dst=e4:f8:79:ed:0d:c3 actions=output:"s2-eth1"
cookie=0x0, duration=273.322s, table=0, n_packets=10, n_bytes=756, priority=3, in_port="s2-eth1", dl_dst=e6:1b:6a:a2:38:2e actions=output:"s2-eth3"
cookie=0x0, duration=273.321s, table=0, n_packets=7, n_bytes=574, priority=3, in_port="s2-eth2", dl_dst=e6:1b:6a:a2:38:2e actions=output:"s2-eth3"
h3"
```

3. Below is the image where h1, h2, h3 ping the 10.0.0.42 server.

```
(env) ubuntu@ip-172-31-1-0:~/col724$ sudo python3.8 topology.py
mininet> h1 ping 10.0.0.42 -c 1
PING 10.0.0.42 (10.0.0.42) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable

--- 10.0.0.42 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

mininet> h2 ping 10.0.0.42 -c 1
PING 10.0.0.42 (10.0.0.42) 56(84) bytes of data.
From 10.0.0.2 icmp_seq=1 Destination Host Unreachable

--- 10.0.0.42 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

mininet> h3 ping 10.0.0.42 -c 1
PING 10.0.0.42 (10.0.0.42) 56(84) bytes of data.
From 10.0.0.3 icmp_seq=1 Destination Host Unreachable

--- 10.0.0.42 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

mininet>
```

To implement load balancing considering server load:

1. Monitor server performance metrics.
2. Define load balancing criteria (e.g., CPU usage, latency).
3. Implement dynamic load balancing logic.
4. Set load thresholds and auto-scaling policies.
5. Include fallback mechanisms for overloaded servers.